

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re U.S. Patent Application of)
)
KABASAWA et al.)
)
Application Number: To be Assigned)
)
Filed: Concurrently Herewith)
)
For: INFORMATION PROCESSING DEVICE FOR)
MULTIPLE INSTRUCTION SETS WITH)
RECONFIGURABLE MECHANISM)
)
ATTORNEY DOCKET NO. HITA.0406)

Honorable Assistant Commissioner
for Patents
Washington, D.C. 20231

**REQUEST FOR PRIORITY
UNDER 35 U.S.C. § 119
AND THE INTERNATIONAL CONVENTION**

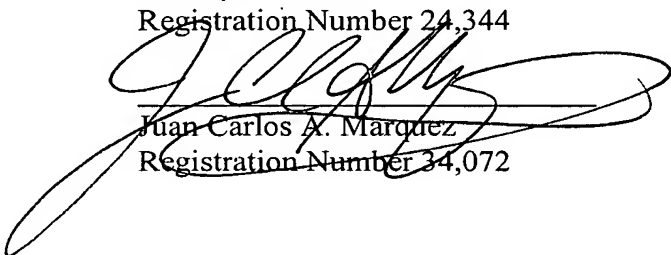
Sir:

In the matter of the above-captioned application for a United States patent, notice is hereby given that the Applicant claims the priority date of August 2, 2002, the filing date of the corresponding Japanese patent application 2002-225487.

A certified copy of Japanese patent application 2002-225487, is being submitted herewith. Acknowledgment of receipt of the certified copy is respectfully requested in due course.

Respectfully submitted,

Stanley P. Fisher
Registration Number 24,344



Juan Carlos A. Marquez
Registration Number 34,072

REED SMITH LLP
3110 Fairview Park Drive
Suite 1400
Falls Church, Virginia 22042
(703) 641-4200
June 30, 2003

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2002年 8月 2日

出 願 番 号

Application Number:

特願2002-225487

[ST.10/C]:

[JP2002-225487]

出 願 人

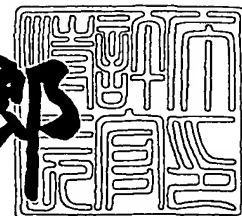
Applicant(s):

株式会社日立製作所

2003年 4月25日

特 許 庁 長 官
Commissioner,
Japan Patent Office

太田信一郎



出証番号 出証特2003-3030675

【書類名】 特許願

【整理番号】 H02005421A

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 9/45.

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所中央研究所内

【氏名】 樺沢 正之

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所中央研究所内

【氏名】 入江 直彦

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所中央研究所内

【氏名】 津野田 賢伸

【発明者】

【住所又は居所】 東京都小平市上水本町五丁目 2 0 番 1 号 株式会社日立製作所半導体グループ内

【氏名】 入田 隆宏

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所中央研究所内

【氏名】 十山 圭介

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所中央研究所内

【氏名】 山田 哲也

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社 日立製作所

【代理人】

【識別番号】 100075096

【弁理士】

【氏名又は名称】 作田 康夫

【電話番号】 03-3212-1111

【手数料の表示】

【予納台帳番号】 013088

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報処理装置

【特許請求の範囲】

【請求項 1】

情報処理装置であって、

第 1 命令セットを固有命令として実行するための命令実行部と、

第 2 命令セットのうち第 1 命令群に属する命令を前記第 1 命令セットに変換して前記命令実行部に供給するための命令変換回路とを備え、

前記情報処理装置は、前記第 2 命令セットのうち前記命令変換回路で変換しない第 2 命令群に属する命令が入力された場合には、ソフトウェアによって前記第 1 命令セットに変換して前記命令実行部で実行し、

前記命令変換回路は、前記第 2 命令セットのうち前記第 1 命令群と前記第 2 命令群の何れかに属するかを再定義可能に記憶するための第 1 記憶領域を有する情報処理装置。

【請求項 2】

請求項 1 において、

前記第 2 命令セットに属する命令群は、複数の命令カテゴリに分類され、

前記第 1 記憶領域は、前記複数の命令カテゴリ毎に、前記第 1 命令群と前記第 2 命令群の何れかに属するかを記憶する情報処理装置。

【請求項 3】

請求項 2 において、前記複数の命令カテゴリには、ローカル変数アクセス命令群、配列アクセス命令群、32ビット演算命令群、64ビット演算命令群、浮動小数点命令群、スタック操作命令群、サブルーチンジャンプ／リターン命令群、フロー制御命令群の何れか一つが含まれる情報処理装置。

【請求項 4】

請求項 1 において、前記命令変換回路は、前記第 2 命令セットが入力されると前記第 1 記憶領域の情報を参照して、前記第 1 命令群に該当する場合には当該命令変換回路に含まれる命令変換テーブルにより前記第 1 命令フォーマットに変換し、前記第 2 命令群に属する場合には前記ソフトウェアに入力された命令の実

行を委ねる情報処理装置。

【請求項 5】

請求項 1 において、前記ソフトウェアは前記第 1 命令セットで記述される情報処理装置。

【請求項 6】

請求項 1 において、

前記命令変換回路は、前記ソフトウェアの第 1 バージョンで定義されたオペコードとその処理を行うための制御回路と、前記ソフトウェアがバージョン変更された際に前記制御回路におけるオペコードとその処理の対応関係を変更するための第 2 記憶領域を有する情報処理装置。

【請求項 7】

請求項 6 において、

前記命令変換回路は、前記第 2 命令セットにおいて定義される定義される配列又はフィールドをメモリ上に配置するための情報を再定義可能に記憶する第 3 記憶領域を有する情報処理装置。

【請求項 8】

請求項 7 において、

前記第 3 記憶領域は、

前記第 2 命令セットにおいて定義される配列をメモリ上に配置する場合のメモリ上の先頭アドレスから前記配列の先頭エントリまでのオフセット値を記憶するための第 1 領域と、

前記配列をメモリ上に配置する場合のメモリ上の先頭アドレスから配列の配列長を格納するフィールドまでのオフセット値を記憶するための第 2 領域と、

前記第 2 命令セットにおいて定義されるフィールドをメモリ上に配置する場合のメモリ上の先頭アドレスから前記フィールドの先頭エントリまでのオフセット値を記憶するための第 3 領域を有する情報処理装置。

【請求項 9】

情報処理装置であって、

第 1 命令セットを固有命令として実行するための命令実行部と、

第 2 命令セットのうち第 1 命令群に属する命令を前記第 1 命令セットに変換して前記命令実行部に供給するための命令変換回路とを備え、

前記情報処理装置は、前記第 2 命令セットのうち前記命令変換回路で変換しない第 2 命令群に属する命令が入力された場合には、ソフトウェアによって前記第 1 命令セットに変換して前記命令実行部で実行され、

前記命令変換回路は、前記ソフトウェアの第 1 バージョンで定義されたオペコードとその処理を記憶する制御回路と、前記ソフトウェアがバージョン変更された際に前記制御回路におけるオペコードとその処理の対応関係を変更するための第 2 記憶領域を有する情報処理装置。

【請求項 1 0】

請求項 9 において、

前記制御回路は、前記ソフトウェアの第 1 バージョンで定義されたオペコードとその処理の対応関係の他に前記ソフトウェアの第 2 バージョンで定義されたオペコードとその処理の対応関係を記憶し、

前記第 2 記憶領域は、前記制御回路が前記第 1 バージョンまたは第 2 バージョンの何れかに対応するかを再定義可能に指定するための領域を有する情報処理装置。

【請求項 1 1】

請求項 9 において、

前記制御回路は、前記ソフトウェアの第 1 バージョンで定義されたオペコードとその処理対応関係を記憶し、

前記第 2 記憶領域は、前記前記制御回路に記憶されるオペコードとその処理の対応関係の一部を無効とすることを再定義可能に指定するための第 1 領域と、

前記第 1 領域によって無効とされた処理に対する新たなオペコードを再定義可能に記憶するための第 2 領域を有する情報処理装置。

【請求項 1 2】

請求項 9 において、

前記第 2 命令セットは J a v a バイトコードであり、

前記ソフトウェアは前記 J a v a バイトコードを前記情報処理装置で実行する

ソフトウェア・バーチャル・マシンであり、

前記第 2 記憶領域は、前記 J a v a バイトコードのうち拡張バイトコードのオペコードとその処理内容の対応関係を再設定可能とする情報処理装置。

【請求項 1 3】

情報処理装置であって、

固有命令を実行するための命令実行部と、

J a v a バイトコードのうちの第 1 命令群に属する命令を前記固有命令に変換して前記命令実行部に供給するための命令変換回路とを備え、

前記情報処理装置は、前記 J a v a バイトコードのうち前記命令変換回路で変換しない第 2 命令群に属する命令が入力された場合には、ソフトウェア・バーチャル・マシンによって前記固有命令に変換して前記命令実行部で実行され、

前記命令変換回路は、前記 J a v a バイトコードの前記第 1 命令群と前記第 2 命令群と区分を再定義するための第 1 記憶領域と、前記 J a v a バイトコードのオペコードとその処理の対応関係を再定義するための第 2 記憶領域と、前記 J a v a バイトコードにおいて定義される配列又はフィールドをメモリ上に配置するための情報を再定義するための第 3 記憶領域を有する情報処理装置。

【請求項 1 4】

請求項 1 3 において、

前記情報処理装置は、前記第 1 から第 3 記憶領域に所定の値を書き込むことにより前記ソフトウェア・バーチャル・マシンのバージョン変更に対応可能とされる情報処理装置。

【請求項 1 5】

請求項 1 4 において、

前記 J a v a バイトコードは、複数の命令カテゴリに分類され、

前記第 1 記憶領域は、前記複数の命令カテゴリ毎に、前記第 1 命令群と前記第 2 命令群の何れかに属するかを記憶する情報処理装置。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は情報処理装置、特に中間言語を高速に実行するための情報処理装置における中間言語のバージョンの互換性に関する。

【 0 0 0 2 】

【従来の技術】

近年、J a v a 言語で記述されたアプリケーションが携帯電話や携帯端末で急速に広がっている。J a v a とはサン・マイクロシステムズ社により開発された C ++ に類似したオブジェクト指向プログラミング言語である。なお、” J a v a ” はサン・マイクロシステムズ社の登録商標である。J a v a 言語が受けられる要因として、J a v a 言語で記述されたプログラムは、中間言語の形式で配布され仮想マシンを用いてマシン固有の C P U 命令（固有命令）に変換して実行される特徴が挙げられる。実行時において仮想マシンを備える C P U であれば、C P U の種類に依存せずに J a v a アプリケーションを実行することができるので移植性が高い。なお、中間言語とは、J a v a の実行オブジェクトを生成するためのコンパイルの結果であり、J a v a バイトコードまたは省略してバイトコードとも呼ばれる。

【 0 0 0 3 】

仮想マシン（以下” V M ” と呼ぶ）は、一般にソフトウェア（これを以下” ソフト V M ” と呼ぶ）で提供されるが、各バイトコードをインタプリタで解釈実行するために一般には低速である。このため、使用頻度の高いバイトコードをハードウェア実行することで高速化を行なうハードウェアアクセラレータが知られている。ハードウェアアクセラレータの例については「日経エレクトロニクス」，no. 797, pp. 168-176, 2001.6.4（この記事は Java to go : Part 1; Microprocessor Report, vol. 15, no. 2, Feb. 2001 の翻訳である）に記載される。

【 0 0 0 4 】

、【発明が解決しようとする課題】

本願発明者等は、本願に先だってハードウェアアクセラレータでバイトコードを処理する際の V M のバージョンアップについて検討を行った。即ち、ハードウェアアクセラレータは一般には特定のバージョンの V M を前提としてその V M で定義される全部又は一部の命令をハードウェアで実行することにより高速化を図

るものである。従って、一つのハードウェアアクセラレータを作成した後、VMのバージョンが変更されるとハードウェアが正しく動作しなくなる場合が生ずる。これに対応するためにVMのバージョンアップに合わせてハードウェアを変更するとハードウェアの開発効率が良くない。

【 0 0 0 5 】

VMの仕様はサン・マイクロシステムズ社により決定される。サン・マイクロシステムズ社はJ a v a 2として複数の仕様を公開しており、そのうち組み込み機器向けの仕様はJ 2 M Eである。この仕様にはさらに、ネットワーク情報機器用のC D Cと、C P Uやメモリに制限のある携帯型ネットワーク情報機器用のC L D Cの2つがある。両者の基本的なバイトコードのセットは同一である。

【 0 0 0 6 】

基本バイトコードとは別に、J a v a V Mの実装依存の仕様として、拡張バイトコードがある。これは基本バイトコードの割り当てであまっているコード空間に新たに拡張バイトコードを割り当てて、J a v a 実行中に、複雑なバイトコードを単純な拡張バイトコードに置き換えることが認められている。拡張バイトコードはVMの仕様として定められているのではなく、独自に処理内容を決められるが、S u nが示す参照実装が事実上の標準となっている。C D CとC L D Cでは拡張バイトコードのセットが異なる。さらに、それぞれの参照実装のマイナーバージョンアップでも、拡張バイトコードの処理内容が変わったり、同じ処理をするバイトコードでもオペコードの割り当てが変わってしまう。これらのVMの差異はソフトウェアインタプリタでは容易に改造できるが、ハードウェアで拡張バイトコードの実行をサポートしようとする、VMがバージョンアップする毎に新たなハードウェアを作らなければ対応できない。これではハードウェア開発の効率が良くないことに本願発明者等は気がついた。このほか、J a v aの仕様にあるフィールド、配列の構造がVMのバージョンによって異なる場合があり、VMにあわせてハードウェアを作成するのも効率が悪い。

【 0 0 0 7 】

本発明の目的は、VMによって定義される命令をハードウェアで実行する際にVMのバージョンに柔軟に対応することができる情報処理装置を提供することに

ある。

【 0 0 0 8 】

【課題を解決するための手段】

本願において開示される発明のうち代表的なものの概要を簡単に説明すれば、下記の通りである。すなわち、第1命令セットを固有命令として実行するための命令実行部と、第2命令セットのうち第1命令群に属する命令を前記第1命令セットに変換して前記命令実行部に供給するための命令変換回路とを備え、前記情報処理装置は、前記第2命令セットのうち前記命令変換回路で変換しない第2命令群に属する命令が入力された場合には、ソフトウェアによって前記第1命令セットに変換して前記命令実行部で実行し、前記命令変換回路は、前記第2命令セットのうち前記第1命令群と前記第2命令群の何れかに属するかを再定義可能に記憶するための第1記憶領域を有するよう情報処理装置を構成する。

【 0 0 0 9 】

更に望ましくは、前記命令変換回路は、前記ソフトウェアの第1バージョンで定義されたオペコードとその処理を行うための制御回路と、前記ソフトウェアがバージョン変更された際に前記制御回路におけるオペコードとその処理の対応関係を変更するための第2記憶領域を第1記憶領域に追加して又は代替して設けるようすると良い。

【 0 0 1 0 】

更に望ましくは、前記命令変換回路は、前記第2命令セットにおいて定義される定義される配列やフィールドをメモリ上に配置するための情報を再定義可能に記憶する第3記憶領域を第1記憶領域に追加して又は代替して設けるようすると良い。

【 0 0 1 1 】

【発明の実施の形態】

以下、本発明に係る情報処理装置及びシステムの好適な実施の形態について、添付図面を参照しながら説明する。特に制限されないが、実施例の各ブロックを構成する回路素子は、特に制限されないが、公知のCMOS（相補型MOSトランジスタ）やバイポーラトランジスタ等の半導体集積回路技術によって、単結晶

シリコンのような1個の半導体基板上に形成される。

(1) システムの構成

図1に、本発明の情報処理システムの実施例を示す。図1のプロセッサチップCPU4は、一般的なプロセッサと同様、Javaバイトコードとは異なるプロセッサ固有の命令セット（固有命令）を持っている。特に制限されないが、この実施例のCPU4には、フェッチ部（フェッチ回路）41、命令変換部（命令変換回路）2、デコード部（命令デコード回路）42、命令実行部（命令実行回路）43が、1個の半導体基板上に形成されている。メモリ装置3は、Javaバイトコード、ソフトVM、フレーム等Javaを実行するための情報を保持している。更に、メモリ装置3は、固有命令で記述されたアプリケーションソフトウェアやソフトウェアの実行のためのワーク領域なども含む。尚、メモリ装置3は、キャッシュメモリのようなSRAM、メインメモリとなるべきDRAM、又はそのキャッシュメモリとメインメモリの両方を持つものを想定しており、各々がCPU4と同一チップであっても別々のチップであっても構わない。

【0012】

メモリ装置3から通常のCPU4の固有命令が供給された場合は、命令変換部2を介さない経路（41→42→43）で実行される。一方、メモリ装置3からバイトコードが供給された場合は、フェッチ部41を通り、命令変換部2にバイトコードが入力され、入力されたバイトコードは固有命令列に変換され、デコード部42、実行部43に送られ実行される。

【0013】

CPU4に変換回路2が命令変換を行うサポートバイトコード（第1命令群）が入力された場合、バイトコード変換テーブル21で、Javaバイトコードから1の固有命令又は固有命令列に変換される。変換回路2が命令変換を行わずソフトVMで命令変換を行う非サポートコード（第2命令群）が入力された場合、制御回路22で検知し、バイトコード変換テーブル21からソフトVMへ切り替えるための命令列が出力され、この命令列が実行されてソフトVMへ処理が移る。

【0014】

以上のように、本願発明ではサポートバイトコードと非サポートバイトコード

に応じて、命令変換回路とソフトVMが選択的に処理を行う。ここで、命令変換回路は、ハードウェアであるために、特定のバージョンのVMを前提として設計される。ところがVMのバージョンが変更されると、定義の変更のために変更されたバージョンのVMでコンパイルしたバイトコードは、古いバージョンのVMで作り込んでしまった命令変換回路では所望の動作をしない場合が生ずる。そこで、本願発明者等はVMのバージョン変更の内容について検討した結果、過去及び将来のVMの変更に対応するには以下詳述するパラメータを再定義可能に記憶する設定レジスタ1によって対応することを見いだした。

【0015】

即ち、この設定レジスタ1の値を再設定することによりVMのバージョン変更に対応できることになる。典型的にはVMバージョン1を前提として命令変換回路を作成した場合には、そのバージョン1に対応するソフトVMと設定レジスタ1の第1パラメータテーブルをメモリ装置3に記憶する。ここでCPU2に電源が投入されて初期化を行う際に第1パラメータがメモリ装置から設定レジスタ1にロードされ命令変換回路の初期設定がなされる。このようにして命令変換回路はバージョン1のソフトVMとセットになって動作する。尚、第1パラメータは特別な変更を指示しなければ設定レジスタ1にデフォルト値として初期値で設定するようにしても良い。

【0016】

次にVMのバージョンが変更となりバージョン2になった場合には、バージョン1に代えてバージョン2のソフトVMとそれに対をなす第2パラメータテーブルをメモリ装置に記憶する。この場合は命令変換回路はバージョン2のソフトVMとセットになってバージョン2のVMでも動作させることができるようになる。

【0017】

更に本願発明では各々バージョン1とバージョン2の両方でコンパイルされた2つのバイトコードアプリケーションが存在している場合でも、各バージョンのソフトVMとパラメータテーブルを対として切り替えることで実行可能となる。

【0018】

ここで、設定レジスタ1は少なくとも実行バイトコード指定レジスタ（第1記憶領域）11を含み、更に拡張バイトコード指定レジスタ（第2記憶領域）12を含むと好ましく、更に配列／フィールドオフセットレジスタ（第3記憶領域）13を含むと好ましい。以下それぞれのレジスタの役わりを詳細に示す。

（2）サポートバイトコードの指定方法

レジスタが実行バイトコード指定レジスタ11は、ハードウェアで実行するバイトコードを指定するレジスタである。バイトコードごとに、全オペコード分のレジスタを用意する方法と、カテゴリ毎にレジスタを用意する方法が考えられる。ここでは、特に制限されないがレジスタサイズが小さくて済む後者の方式を好ましい実施例として示す。

【0019】

図2に実行バイトコード指定レジスタ11の詳細を示す。本願では約230のバイトコードをとくに制限されないがローカル変数アクセス命令群、配列アクセス命令群、32ビット演算命令群、64ビット演算命令群、浮動小数点命令群、スタック操作命令群、サブルーチンジャンプ／リターン命令群、フロー制御命令群などを含む14のカテゴリに分類した。それぞれのカテゴリに1つまたは複数のバイトコードが含まれる。カテゴリ1実行指定レジスタCAT1_REGは1番目のカテゴリのバイトコードをサポート命令として扱うか、非サポート命令として扱うかを"1"又は"0"で指定する1ビットのレジスタである。他のカテゴリについても同様にCAT2_REGからCAT14_REGで指定する。

【0020】

図3にバイトコード指定レジスタ11の設定と、バイトコードがハードウェアで実行されるかソフトVMで実行されるかの例を表で示す。バイトコードBCとしてBC1、BC2、...は、それぞれカテゴリ1CAT1、カテゴリ2CAT2...に分類されていたと考える。この場合第1パラメータテーブルPARA1を実行バイトコード指定レジスタ11に設定した場合（CAT1_REG=1，CAT2_REG=1，CAT3_REG=0,...）には、図3の例示範囲ではBC5がSの記号で示すようにソフトVMで実行される。一方、BC1からBC4は、Hの記号で示すようにハードウェアで実行されるように設定される。

【0021】

また、第2パラメータテーブルP A R A 2を実行バイトコード指定レジスタ11に設定した場合 (CAT1_REG=1, CAT2_REG=0, CAT3_REG=1,...)には、図3の例示範囲ではB C 2とB C 4がソフトVMで実行され、B C 1、B C 3、B C 5はハードウェアで実行されるように設定される。

【0022】

実行バイトコード指定レジスタ11に設定する典型例は以下の通りである。プロセッサがシステムに搭載後にVMのバージョンが変わり、例えば配列の構造が変わった場合、配列アクセスに関わるバイトコードが属するカテゴリのカテゴリ実行指定ビットを非サポート命令として扱うように設定する。このように設定すると、制御回路22が配列アクセスのバイトコードを非サポートバイトコードとして検知し、バイトコード変換テーブル21がソフトVMに切り替えるための命令列を発行し、ソフトVMへ操作が移ることで、このバイトコードはソフトVMで実行される。このように、VMのバージョンの差異により配列の構造が合わなくなった場合には、非サポート命令として扱うことで、配列構造の異なったVMにも対応することができる。その他、浮動小数の表現方法がVMのバージョンで異なったり、エンディアンの仕様で64ビット演算が行えなくなった場合なども、該当するカテゴリ実行指定ビットを非サポート命令として扱うように設定し、ソフトVMに処理を渡すことで対応することができる。

【0023】

本実施例ではカテゴリ毎にレジスタを用意する方法を実施例としてしめたが、それに限定されることはない。即ち、J a v aバイトコードはオペコードの数は約230程度であり、1つのオペコードについて1ビットのレジスタを割り当てても230ものラッチレジスタを要することとなるため命令変換回路の回路規模を増大してしまい、カテゴリ毎に記憶すると回路規模が低減できる。しかし、一般にオペコードの数が少ない命令セットを扱う場合には前オペコード分にレジスタを方法をとっても回路規模は大きな問題とはならない。

(3) 拡張バイトコードの指定方法

図5に、拡張バイトコードの処理内容やオペコードの番号の割り当てがVMの

バージョンによって異なる様子を示す。まず拡張バイトコードとは、J a v a において `iload` や `iadd` 等の基本命令の他に、基本命令を高速に実行するため、定義されている `getfield_quick` 等の VM に特有の拡張命令を指定するバイトコードである。図 5 において、まず処理内容が異なる例として、拡張バイトコードの一つ `EXBC 5` はバージョン 1 の VM 1 ではオペコード 2 3 4 に割り当てられていたとする。これに対して、バージョン 2 の VM 2 では拡張バイトコード `EXBC 5` はサポートされないため `NON` で示している。次にオペコードの番号の割り当てがバージョンにより異なる例を示す。即ち拡張バイトコード `EXBC 1` の処理は、VM 1 では 2 3 0 番に割り当てられるのに対して、VM 2 では 2 3 5 番に割り当てられている。このバージョンの差を解消するためのレジスタが、拡張バイトコードマップレジスタ 12 である。

【 0 0 2 4 】

図 4 に拡張バイトコード指定レジスタ 12 の詳細を示す。このレジスタ 1 2 は、特定 VM バージョン指定レジスタ `VM_VER_REG`、オペコードオーバーライドインーブル指定レジスタ `OVREN_REG`、オペランドフォーマット指定レジスタ `OPFM_REG`、拡張バイトコード指定レジスタ `EXBC1_REG`, `EXBC2_REG`, `EXBC3_REG`, ... を含む。

【 0 0 2 5 】

特定 VM バージョン指定レジスタ `VM_VER_REG` は、例えばバージョン 1 とバージョン 2 の VM が既に存在してる場合に "0" によりバージョン 1 を指定し、"1" によりバージョン 2 を指定するものである。更に多くのバージョンが存在する場合にはこの `VM_VER_REG` のビット幅を拡張することでバージョンの指定をすればよい。ここで、制御回路 22 は、製造時に確定しているバージョン 1 とバージョン 2 の両方の VM に対する拡張命令のオペコードと処理を予め記憶してものとする。従って、特定 VM バージョン指定レジスタ `VM_VER_REG` は、制御回路 22 に対してバージョン 1 とバージョン 2 のいずれの対応関係の処理を行うかを指定するスイッチとして働く。

【 0 0 2 6 】

これに対して、プロセッサがシステムに搭載後に VM のバージョンが変わり、

想定していた拡張バイトコードのオペコードが合わなくなった場合、オペコードオーバーライドイネーブル指定レジスタOVREN_REGを、イネーブルに設定にし、拡張バイトコード指定レジスタEXBC1_REGには、拡張バイトコードEXBC1の処理に合ったオペコードを設定する。例えば図5の例で、EXBC1はバージョン1では230、バージョン2では235である。このバージョン1と2の切り替えは、上述の通りVMバージョン指定レジスタVM_VER_REGで変更できる。これに対して、バージョン3で更に変更されEXBC1に240が割り当てられた場合にOVREN_REGを"1"に設定し、EXBC1_REGに"240"を設定する。他の拡張バイトコード指定レジスタEXBC2_REG, EXBC3_REG, ..も同様に指定する。命令変換部2に入力されたバイトコードが拡張バイトコード指定レジスタに設定されたオペコードと一致するかを制御回路22が検知し、一致すると対応する命令列をバイトコード変換テーブル21が出力し、一致しない場合は非サポート処理に移るための命令列を出力して、ソフトVMに処理を渡す。

【0027】

オペランドフォーマット指定レジスタOPFM_REGは、フィールド系拡張バイトコードのオペランドフォーマットの指定方法を変更するために使用する。フィールドにアクセスするバイトコードのオペランドは、バージョンに応じて変更になる場合があることに対応するためである。フィールド系拡張命令は、オペコードが1バイトとそれに連続する2バイトのオペランドの合計3バイトで指定される。しかし、後ろ2バイトのオペランドは(1)第1オペランド有効/第2オペランド有効、(2)第1オペランド有効/第2オペランド無効、(3)第1オペランド無効/第2オペランド有効となるような変更が想定される。オペランドフォーマット指定レジスタOPFM_REGに"00"、"01"、"10"のような値を指定することにより(1)~(3)のようなオペランドフォーマットの変更が可能とされる。尚、OPFM_REGは、OVREN_REGの設定がイネーブルとされるときに設定を必要とし、デフォルト値でを利用する場合には設定を必要としない。

【0028】

以上の通り、拡張バイトコードの詳細なバージョン変更に対応する手段を示したが、さらに拡張バイトコードがVMのバージョンアップにより上記の方法では

対応できなくなった場合は、実行バイトコード指定レジスタ11を非サポート命令に設定し、ソフトVMで対応する。従って、VMのバージョン変更の吸収という点に着目すれば、実行バイトコード指定レジスタ11のほうがより本質的である。実行バイトコード指定レジスタ11の意義は、更に、ハードウェア性能の維持と言う点にある。

(4) 配列／フィールドオフセットの指定方法

配列やフィールドの構造がVMのバージョンにより違う構造である場合がある。現状のVMは配列の先頭にヘッダ部分があり、ヘッダ部分のオフセットを足したアドレスでデータにアクセスする。今後、ヘッダ部分HEADERのオフセットがVMによって変更される事もあり得ると考え、配列／フィールドオフセットレジスタ13には図6に示すように、配列先頭オフセットレジスタATOP_REGを設けて、オフセットを変更できる仕様とした。また、配列長LENは配列のヘッダ部分の中にあり、配列長のフィールドまでのオフセットを配列長フィールドオフセットレジスタALEN_REGで設定する。これらの設定すると、バイトコード変換テーブル21が、その設定に対応した命令列を出力する。フィールドに対しても同様で、フィールド先頭オフセットレジスタFTOP_REGを設けて、特殊命令テーブル23から対応した命令列を発行する。なお、Javaにおける配列とフィールドの概念図を図7に示す。図において(A)が配列オフセット指定レジスタ、(B)が配列長フィールドオフセット指定レジスタ、(C)がフィールドオフセット指定レジスタのそれぞれの設定内容を示している。

【0029】

さらにVMのバージョンアップにより上記の方法では対応できなくなった場合は、実行バイトコード指定レジスタ11を非サポート命令に設定し、ソフトVMで対応する。従って、VMのバージョン変更の吸収という点に着目すれば、実行バイトコード指定レジスタ11のほうがより本質的である。配列／フィールドオフセットレジスタ13の意義は、更に、ハードウェア性能の維持と言う点にある。

【0030】

以上サポートバイトコードの指定方法、拡張バイトコードの指定方法、配列／フィールドオフセットの指定方法に分類し対応するレジスタ11～13を例示し

て示した。レジスタ 11～13 の分類及びその中の詳細なレジスタのまとまりは便宜上のものであり、結果的に本願のレジスタに対応する情報の記憶領域があれば本願の目的は達成できるのでそのような改変を行ってもよい。ここで示したレジスタは、揮発性の読み書き可能なラッチを想定しているが、一般には再定義可能な記憶領域があればよい。

【 0 0 3 1 】

図8は本発明の第2の実施例である。本実施例はCPU4の外部に命令変換部2がありこの命令変換部が固有命令を実行するCPUとは独立の半導体チップに形成されている点が図1の実施例とは異なる。命令変換部が別チップなので既存のCPUチップに適用できる利点がある。設定レジスタ1は、第1の実施例と同様であり、図1の実施例と同様にVMのバージョンの違いに対応できる。

【 0 0 3 2 】

図9に本発明のの好適な応用例としての携帯情報システムの構成図を示す。より具体的には携帯電話の構成の一例である。大きく分けて、通信部分とアプリケーション部分から構成される。通信部分は、電波を送受信するアンテナ（RF）部83、ベースバンドのモデム、コーデックを行うベースバンドプロセッサ（BASEBAND）部81、主記憶（MEM）82である。アプリケーション部分は、本発明の実施例のバイトコードアクセラレータ10を備えるCPU1を内蔵するマイクロプロセッサ（PROCESSOR）70において、インターフェース（I/F）73を介してアプリケーションプロセッサとベースバンドプロセッサが接続され、周辺インターフェース（PERIPHERAL）74を介してカメラ（CMR）75、メモ리카ード（CARD）76、音源IC（SOD）77、キー（KEY）78が接続され、外部バスを介して液晶（LCD）79、主記憶（MEM）80が接続される。本システム構成例は、携帯電話向けであったが、携帯情報端末やデジタルカメラなど多様なシステム構成例が考えられる。

【 0 0 3 3 】

本システム構成例では、例えば、次のようにメモリが使用される。Javaアプリケーションは、外部サーバーよりアンテナ83を介して配給され、ベースバンドプロセッサ81を使用し、主記憶82に格納される。ソフトVMは主記憶8

0 か主記憶 8 2 のどちらに配置してもよい。ソフト VM のインタプリタ部はアクセス頻度が高いため、内蔵メモリ上に配置されることが望ましい。この発明により、バージョンの異なる VM でコンパイルされた J a v a アプリケーションがダウンロードされた場合でもハードウェアアクセラレータを利用して高速に実行できるため、携帯情報システムの付加価値を高めることができる。

【 0 0 3 4 】

以上の本願発明は J a v a を例にして説明したが、それに限られることはなく、同様の中間言語及び VM を固有命令を命令セットで実行するプロセッサに対するハードウェアアクセラレータに適用できる。

【 0 0 3 5 】

【発明の効果】

上記手法により、異なる VM でも、設定レジスタの内容を設定することにより、同一ハードウェアで命令変換することができる。また、VM がバージョンアップされた場合でも、ハードウェアを作り直すことなく同一のハードウェアで命令変換を行うことが可能である。

【図面の簡単な説明】

【図 1】

本発明によるプロセッサの概略ブロック図。

【図 2】

実行バイトコード指定レジスタのブロック図。

【図 3】

実行バイトコード指定レジスタの設定例。

【図 4】

拡張バイコードマップレジスタのブロック図。

【図 5】

VM バージョンの違いによる拡張バイトコードの割り当ての違いの例。

【図 6】

配列/フィールドオフセットレジスタのブロック図。

【図 7】

配列/フィールドの構造。

【図 8】

本発明によるアクセラレータの概略ブロック図。

【図 9】

本発明による携帯システムの概略ブロック図。

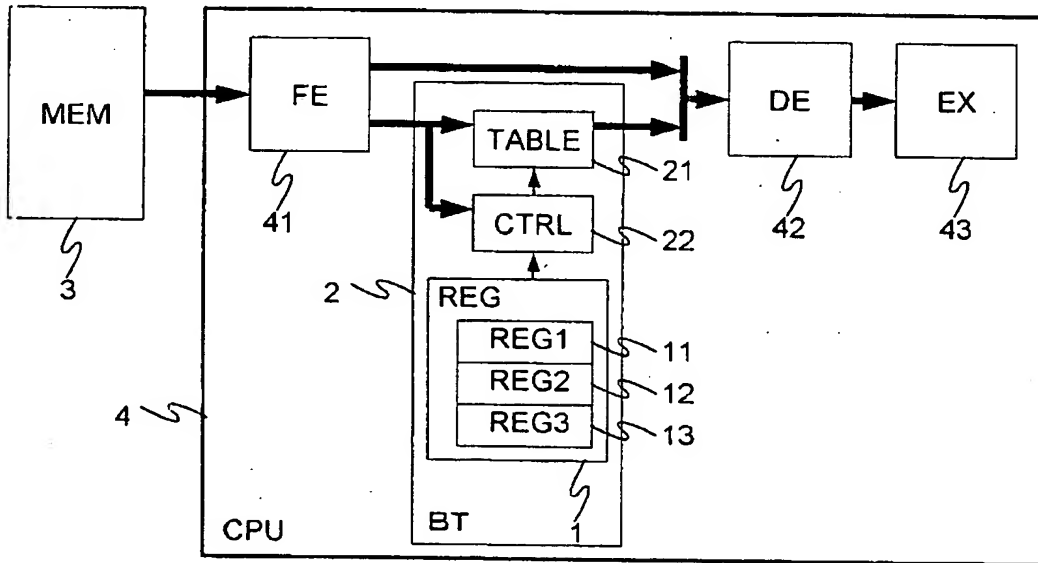
【符号の説明】

- 1 … 設定レジスタ (REG)
- 2 … 命令変換部 (BT)
- 3 … メモリ装置 (MEM)
- 4 … プロセッサ (CPU)
- 1 1 … 実行バイトコード指定レジスタ (REG 1)
- 1 2 … 拡張バイトコード指定レジスタ (REG 2)
- 1 3 … 配列/フィールドオフセットレジスタ (REG 3)
- 2 1 … バイトコード変換テーブル (TABLE)
- 2 2 … 制御回路 (CTRL)
- 4 1 … フェッチ部 (FE)
- 4 2 … デコード部 (DE)
- 4 3 … 実行部 (EX)。

【書類名】 図面

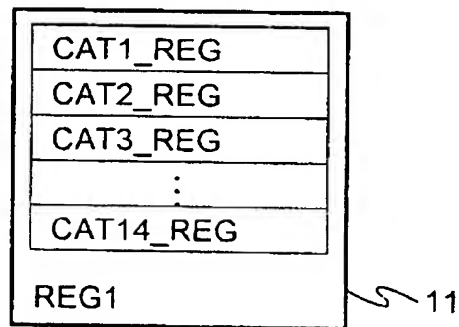
【図 1】

図 1



【図 2】

図 2



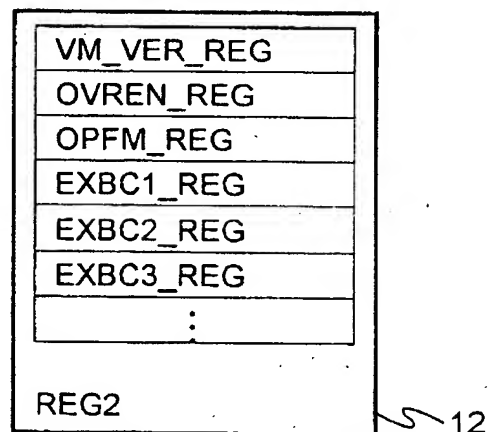
【図3】

図3

BC	CAT	PARA1	PARA2
		CAT1_REG=1 CAT2_REG=1 CAT3_REG=0 ⋮	CAT1_REG=1 CAT2_REG=0 CAT3_REG=1 ⋮
BC1	CAT1	H	H
BC2	CAT2	H	S
BC3	CAT1	H	H
BC4	CAT2	H	S
BC5	CAT3	S	H
⋮	⋮	⋮	⋮

【図4】

図4



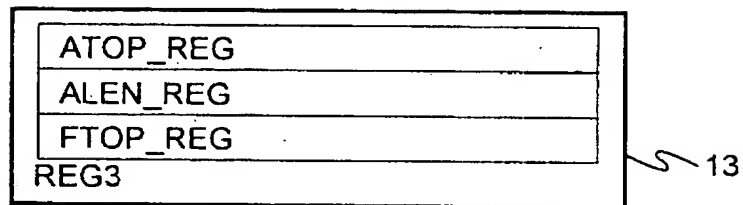
【図5】

図5

	VM1	VM2
EXBC1	230	235
EXBC2	231	237
EXBC3	232	236
EXBC4	233	238
EXBC5	234	NON

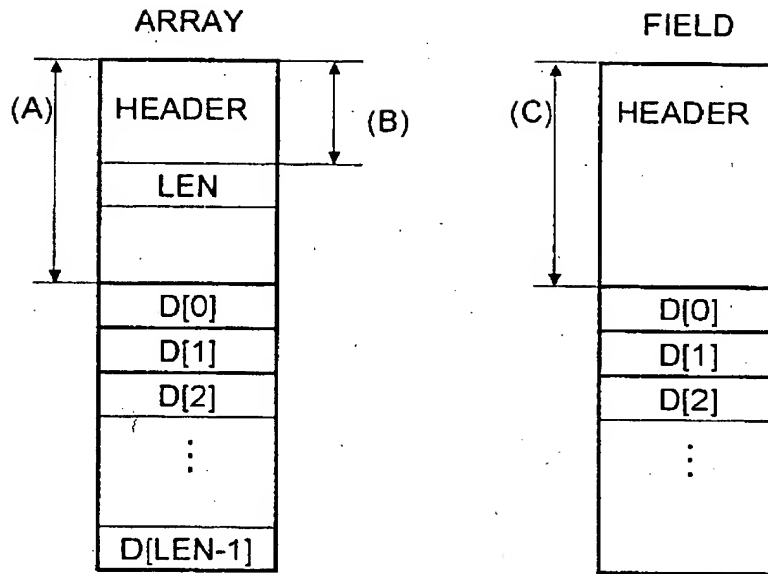
【図6】

図6



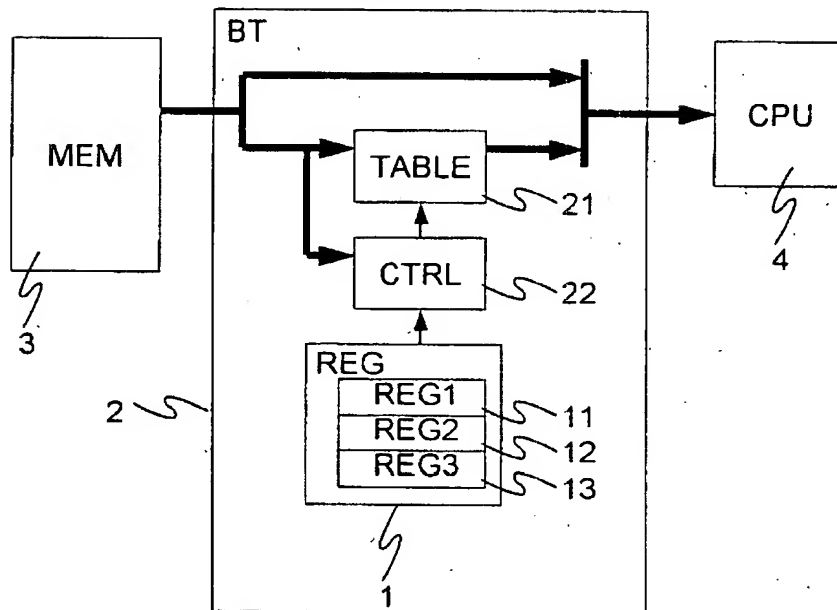
【図7】

図7



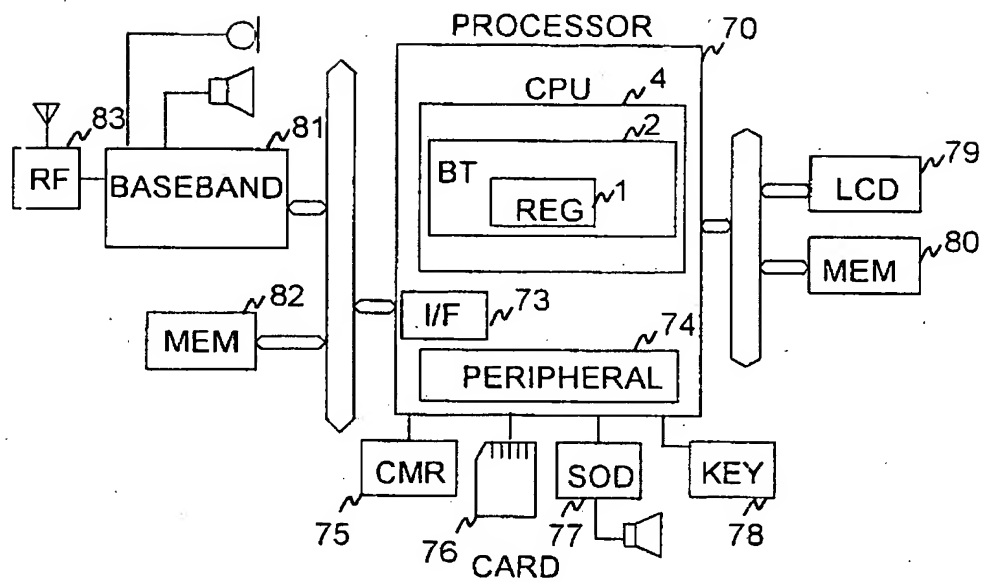
【図8】

図8



【図9】

図9



【書類名】 要約書

【要約】

【課題】 本発明の目的は、中間コード（J a v a バイトコード）を命令実行部が解釈可能な命令列に変換する命令変換回路において、仮想計算機（VM）のバージョンに対応できる機構を提供することにある。

【解決手段】 中間コードのうち第1命令群に属する命令はハードウェアにより解釈可能な命令に変換し、第2命令群に属する命令はソフトウェアによって変換する。命令変換回路は、中間コードが第1命令群と第2命令群のいずれに属するかを再定義可能に記憶するための記憶領域を有するよう情報処理装置を構成する。

【効果】 バージョンの違うVMでも、設定レジスタの内容を設定することにより、同一ハードウェアで命令変換を行うことができる。また、VMがバージョンアップされた場合でも、ハードウェアを作り直すことなく同一のハードウェアで命令変換を行うことが可能になる。

【選択図】 図 1

認定・付加情報

特許出願の番号	特願2002-225487
受付番号	50201144263
書類名	特許願
担当官	第七担当上席 0096
作成日	平成14年 8月 5日

<認定情報・付加情報>

【提出日】 平成14年 8月 2日

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日	1990年 8月31日
[変更理由]	新規登録
住 所	東京都千代田区神田駿河台4丁目6番地
氏 名	株式会社日立製作所